Amendments to the Claims

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

Claim 1 (canceled).

Claim 2 (currently amended): The IC of claim ~~1~~ 7, wherein:
the interrupt controller provides for a fast interrupt request (FIQ) and a normal interrupt request (IRQ) to be programmably masked and prioritized before being issued to the core, and wherein said FIQ directs program execution to a last entry in an interrupt vector table.

Claim 3 (currently amended): The IC of claim ~~1~~ 7, wherein:
the interrupt controller centralizes all interrupt handling and includes programmable interrupt masks for independent enabling and disabling each interrupt source, and for globally disabling all interrupts.

Claim 4 (currently amended): The IC of claim ~~1~~ 7, wherein:
the interrupt controller further includes an interrupt vector control for automatically decoding a highest-priority interrupt for presentation of a programmed interrupt vector to the processor core.

Claim 5 (currently amended): The IC of claim ~~1~~ 7, wherein:
the interrupt controller includes a global-disable control bit for use when a critical portion of program code is executing that is independent of any individual interrupt masks and avoids needing to save and restore mask states that would otherwise increase interrupt latency when a global-disable is lifted.

Claim 6: (canceled)

Claim 7 (previously amended): A single integrated circuit (IC), comprising:
a synthesizable ARM-type microcontroller processor core connected to a program code memory that selectively processes either a THUMB or an ARM program execution stream, wherein said THUMB program execution stream is more economical with program code space, and wherein a program-execution interrupt request forces a hardware switch to said ARM program execution stream;

an interrupt controller that receives interrupt requests and provides a programmable combination of said interrupt requests to the core, wherein each said

interrupt request is associated with one of a plurality of interrupt service routines coded in THUMB program code; and

an interrupt service routine preamble shared amongst said plurality of interrupt service routines, said interrupt service routine preamble is coded in ARM program code to cause a switch to THUMB program execution, said interrupt controller causes the execution of said interrupt service routine preamble before an interrupt service routine is executed.

Claim 8: (canceled)

Claim 9 (currently amended): A system that includes a single integrated circuit (IC), comprising:

a synthesizable ARM-type microcontroller processor core connected to a program code memory that selectively processes either a ~~first program execution stream~~ THUMB or ~~a second~~ an ARM program execution stream, wherein said ~~second~~ THUMB program execution stream is more economical with program code space than said ~~first~~ ARM program execution stream, and wherein a program execution interrupt request forces a hardware switch to said ~~first~~ ARM program-execution stream;

an interrupt controller that receives interrupt requests and provides a programmable combination of said interrupt requests to the core, wherein each said interrupt request is associated with one of a plurality of interrupt service routines coded in ~~said second program execution stream~~ THUMB program code; and

an interrupt service routine preamble shared amongst said plurality of interrupt service routines, said interrupt service routine preamble is coded in ~~said first program execution stream~~ ARM program code to cause a hardware switch to ~~said second program execution stream~~ THUMB program execution, said interrupt controller causes the execution of said interrupt service routine preamble before an interrupt service routine is executed.

Claim 10 (previously added): The system of claim 9, wherein:

the interrupt controller provides for a fast interrupt request (FIQ) and a normal interrupt request (IRQ) to be programmably masked and prioritized before being issued to the core, and wherein said FIQ directs program execution to a last entry in an interrupt vector table.

Claim 11 (previously added): The system of claim 9, wherein:

the interrupt controller centralizes all interrupt handling and includes programmable interrupt masks for independent enabling and disabling each interrupt source, and for globally disabling all interrupts.

Claim 12 (previously added): The system of claim 9, wherein:

the interrupt controller further includes an interrupt vector control for automatically decoding a highest-priority interrupt for presentation of a programmed interrupt vector to the processor core.

Claim 13 (previously added): The system of claim 9, wherein:
the interrupt controller includes a global-disable control bit for use when a critical portion of program code is executing that is independent of any individual interrupt masks and avoids needing to save and restore mask states that would otherwise increase interrupt latency when a global-disable is lifted.

Claim 14 (currently amended): A method that makes a single integrated circuit (IC), comprising:
providing a synthesizable ARM-type microcontroller processor core connected to a program code memory that selectively processes either a ~~first program execution stream~~ THUMB or ~~a second~~ an ARM program execution stream, wherein said ~~second~~ THUMB program execution stream is more economical with program code space than said ~~first~~ ARM program execution stream, and wherein a program execution interrupt request forces a hardware switch to said ~~first~~ ARM program-execution stream;

providing an interrupt controller that receives interrupt requests and provides a programmable combination of said interrupt requests to the core, wherein each said interrupt request is associated with one of a plurality of interrupt service routines coded in ~~said second program execution stream~~ THUMB program code; and

providing an interrupt service routine preamble shared amongst said plurality of interrupt service routines, said interrupt service routine preamble is coded in ~~said first program execution stream~~ ARM program code to cause a hardware switch to ~~said second program execution stream~~ THUMB program execution, said interrupt controller causes the execution of said interrupt service routine preamble before an interrupt service routine is executed.

Claim 15 (previously added): The method of claim 14, wherein:
the interrupt controller provides for a fast interrupt request (FIQ) and a normal interrupt request (IRQ) to be programmably masked and prioritized before being issued to the core, and wherein said FIQ directs program execution to a last entry in an interrupt vector table.

Claim 16 (previously added): The method of claim 14, wherein:
the interrupt controller centralizes all interrupt handling and includes programmable interrupt masks for independent enabling and disabling each interrupt source, and for globally disabling all interrupts.

Claim 17 (previously added): The method of claim 14, wherein:

the interrupt controller further includes an interrupt vector control for automatically decoding a highest-priority interrupt for presentation of a programmed interrupt vector to the processor core.

Claim 18 (previously added): The method of claim 14, wherein:
the interrupt controller includes a global-disable control bit for use when a critical portion of program code is executing that is independent of any individual interrupt masks and avoids needing to save and restore mask states that would otherwise increase interrupt latency when a global-disable is lifted.

Claim 19 (currently amended): A method that processes interrupts using a single integrated circuit (IC), comprising:
receiving one or more program execution interrupt requests, wherein each said interrupt request causes a hardware switch in a synthesizable ARM-type microcontroller processor that selectively processes either a ~~first program execution stream~~ THUMB or ~~a second~~ an ARM program execution stream ~~that~~ wherein said THUMB program execution stream is more economical with program code space than said ~~first~~ ARM program execution stream, and wherein said hardware switch forces execution to said ~~first~~ ARM program-execution stream;

providing a programmable combination of said interrupt requests to the core using an interrupt controller, wherein each said interrupt request is associated with one of a plurality of interrupt service routines coded in ~~said second program execution stream~~ THUMB program code; and

executing an interrupt service routine preamble before an interrupt service routine is executed, wherein said interrupt service routine preamble is shared amongst said plurality of interrupt service routines and is coded in ~~said first program execution stream~~ ARM program code to cause a hardware switch to ~~said second program execution stream~~ THUMB program execution.

Claim 20 (previously added): The method of claim 19, wherein:
the interrupt controller provides for a fast interrupt request (FIQ) and a normal interrupt request (IRQ) to be programmably masked and prioritized before being issued to the core, and wherein said FIQ directs program execution to a last entry in an interrupt vector table.

Claim 21 (previously added): The method of claim 19, wherein:
the interrupt controller centralizes all interrupt handling and includes programmable interrupt masks for independent enabling and disabling each interrupt source, and for globally disabling all interrupts.

Claim 22 (previously added): The method of claim 19, wherein:

the interrupt controller further includes an interrupt vector control for automatically decoding a highest-priority interrupt for presentation of a programmed interrupt vector to the processor core.

Claim 23 (previously added): The method of claim 19, wherein:

the interrupt controller includes a global-disable control bit for use when a critical portion of program code is executing that is independent of any individual interrupt masks and avoids needing to save and restore mask states that would otherwise increase interrupt latency when a global-disable is lifted.

Deleted: ¶